

## 1. 처음

하드웨어의 성능이 끊임없이 발전해 오면서 최적화된 알고리즘의 중요성이 많이 떨어졌다는 것은 부정할 수 없는 사실이다. 이미 잘 만들어진 알고리즘들을 모아둔 라이브러리가 널려 있는데 더 좋은 알고리즘을 고안해 내기 위해 노력하는 시간보다는 이미 준비된 라이브러리를 통해 프로그램을 만드는 것이 생산성이 훨씬 좋기 때문이다. 고작 몇 줄만 코딩해도 구현할 수 있는 기능을 몇 십~몇 백메가바이트나 되는 거대한 프로그램으로 만들어내는 넌센스가 용납되는 이유도 하드웨어의 처리 속도가 비만이 된 프로그램을 실행시키기에 충분히 넉넉하기 때문이다.

하지만, 모든 분야가 그렇게 빠른 속도로 발전해 온 것은 아니다. 끊임없이 빨라질 것 같던 CPU의 클럭도 벽에 부딪혀 멀티코어 형태로 전환되는 추세고, 새로 개발되는 프로세서들도 클럭을 끌어올리는 것보다는 효율을 개선하는 방향으로 바뀌고 있다. 특히 배터리 용량을 획기적으로 끌어올릴 수 있는 기술이 개발되기 전까지는 모바일 기기에서는 '효율'의 중요성을 빼 놓을 수 없다.

효율에 대한 사설(辭說)이 길어진 이유는, 이번 기사에서 다룰 H.264 코덱이 탄생하게 된 배경이 발전 속도가 더딘 모바일 기기의 제약이기 때문이다. 모바일 기기에서 동영상을 감상하는데 제약이 되는 것은 크게 두 가지가 있는데 첫 번째가 배터리 수명, 두 번째가 스트리밍 대역폭이다. 이 중 특히 스트리밍 대역폭은 DMB 통신을 이용하는 모바일 기기든, 네스팟과 같은 무선랜 규격을 사용하는 PDA나 소니 PSP와 같은 기기든, 실시간 스트리밍 서비스를 안정적으로 받기 위해서 충분한 회선 대역폭을 확보해야 하지만 기술적으로 어려운 실정이다. 제한된 대역폭 안에서는 화질/음질 저하를 감수하던지, 아니면 소프트웨어적인 개선. 즉, 코덱의 압축률을 개선하는 방법이 있어야만 하는 상황이다.

H.264 코덱은 국제 표준화 기구인 ITU-T와 ISO에서 공동으로 제안한 비디오 압축 기술로서 ITU-T에서 붙인 H.264라는 명칭 외에도 ISO에서 붙인 MPEG4 Part10/AVC 라는 명칭도 사용한다. H.264/AVC (Advanced Video Coding)은 기존 코덱들과 동등한 화질을 구현하는데 필요한 비트레이트가 상당히 개선 되었다고 하는데, MPEG-4에 비해 2 배, MPEG-2에 비해 최대 18배 까지 비트레이트 감소 효과가 있다는 자료가 있다. 이번 기사에서는 실제로 H.264 코덱의 효율이 얼마나 좋은지를 확인해 보도록 하겠다.

## 2. H.264개요

H.264 코덱은 DivX나 Windows Media Video에 비해 아직 인지도가 낮다. 하지만 주위를 둘러보면 최신 기기에서는 H.264를 어렵지 않게 접할 수 있다. HDTV 셋탑박스, DMB 폰, 소니 PSP 등, 디지털 영상물을 스트리밍하는 장치들이 최근 앞다투어 H.264를 표준으로 채택하고 있다는 뉴스들을 어렵지 않게 찾아볼 수 있을 것이다. H.264 코덱이 만들어진지 한참 되었지만 이제 와서 다루는 이유도 바로 DMB폰이나 PSP 등을 PMP 용도로 사용하고자 하는 수요가 늘어났기 때문이다.

사실 동영상 압축 알고리즘이 과거의 프로그래머들을 바보로 만들만큼 한 순간에 크게 개선되었다는 것을 인정하기는 어렵다. 하지만 H.264가 명칭은 MPEG-4 Part10이지만 태생이 MPEG-4와는 전혀 다르기 때문에 기존 MPEG-4 기반 코덱과는 특성이 전혀 다를 것이고, 특정 조건에서는 훨씬 좋은 결과를 보여줄 것이라는 기대는 해 볼만 하다. 그 조건이 무선 네트워크를 통한 스트리밍에 적합한 것이라면 알려진 정보대로 2배의 효율도 나오지 말라는 법은 없다.

필자가 동영상 압축 기술에 대한 전문가가 아니기 때문에 코덱의 원리를 구체적으로 설명하는 지면은 넘어가고 개요만 소개하도록 하겠다. 기본적으로 미리 알아둘 것은 동영상 압축은 앞프레임과 뒤프레임의 차이점만을 기록하는 방식으로 데이터의 양을 줄인다는 기초 개념이다. 화면 전환이 매우 빠른 경우에는 기록해야 할 차이점이 많기 때문에 압축률이 떨어진다는 점 정도까

지 이해하면 되겠다. 코덱의 특성을 결정짓는 알고리즘의 차이는 두 연속된 이미지의 차이를 분석할 때 기준이 되는 블록의 크기, 그 블록이 이동한 경로를 탐색하는 방식, 찾아낸 차이점을 저장하는 방식의 차이이다.

다음은 H.264의 특성이다.

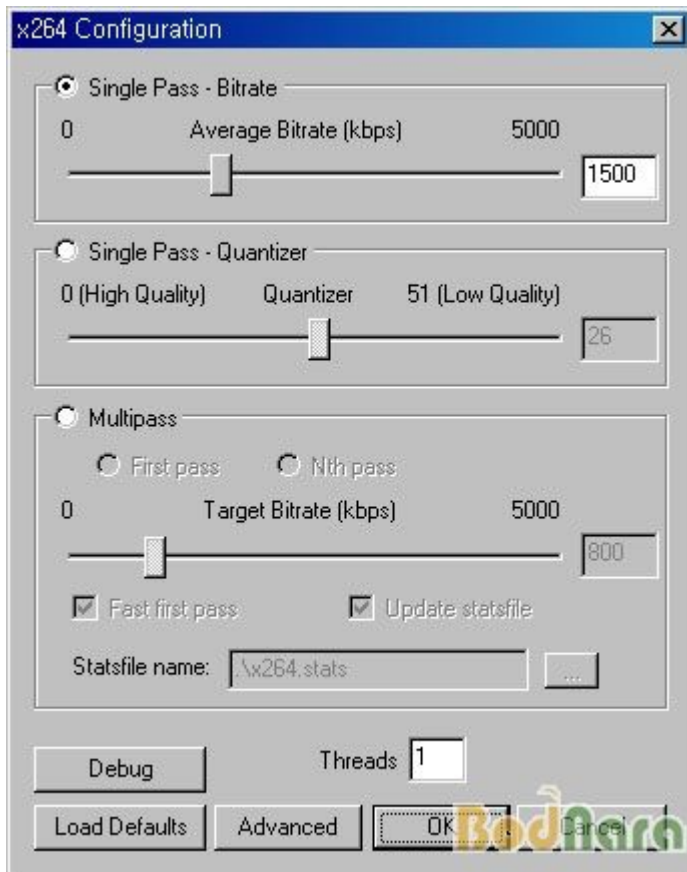
- H.264에서 가장 큰 변화는 프레임간의 차이점을 저장하는 방식의 변화이다. 찾아낸 블록들의 차이를 저장하는 방식으로 코사인 변환(DCT)을 이용하는데, 기존 MPEG-4 Part2에서는 8x8블록 단위로 계산하지만 H.264에서는 4x4 블록 단위로 계산하게 되었다. 더불어 floating point 연산이 필요했던 기존과 달리 이번에는 정수 연산만으로 가능해져 연산오차가 줄어든다는 장점 뿐만 아니라 처리를 전담하는 프로세서의 구조가 단순해질 수 있게 되었다.
- 블록을 탐색하는 단위가 4x4로 바뀌면서 8x8 블록 단위로 탐색할 때보다 보다 정밀한 비교가 가능하다. 또한 4x4 블록 사이즈로 고정된 것이 아니라 필요에 따라 16x16이나 16x8, 8x16, 8x4, 4x8 등과 같이 다양하게 블록크기를 바꿔 적용할 수 있다.
- 기존 동영상 압축 기술은 DCT 변환 이후에 AC/DC 예측이라는 것을 통해 한 프레임 안에 있는 중복된 값을 찾으려 했지만, H.264에서는 DCT 변환 과정 이전에 중복성을 먼저 찾는다.
- H.264에서는 H.263에서 사용했던 루프 필터를 도입하여 낮은 비트레이트에서 영상에 격자무늬가 생기는 것을 막는다. MPEG-4 part2에서는 디코더만 이 기능을 수행했지만, H.264에서는 디코더와 엔코더가 모두 이 기능을 수행하여 영상을 제작할 때의 화질과 완성된 결과물이 재생될 때의 화질 차이가 생기지 않게 하였다.
- 기존에는 바로 앞 프레임과의 차이만을 탐색하여 변화를 기록했지만, H.264에서는 더 이전에 나왔던 프레임과도 비교할 수 있게 되었다. 규칙적인 장면이 반복되는 상황일 때 유리하다.

### 3. 오픈코덱 및 테스트시스템 소개

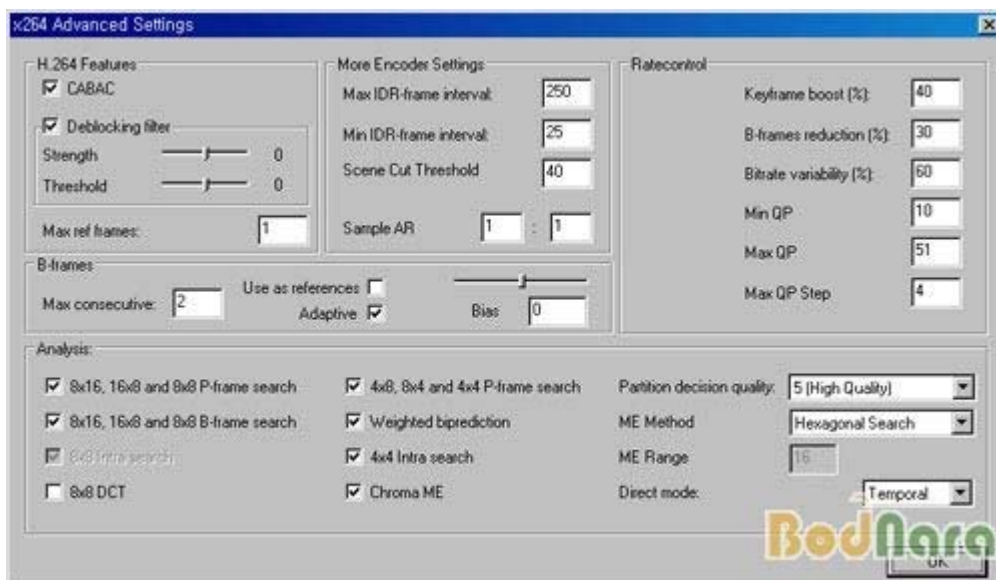
현재까지 나와 있는 H.264 코덱의 종류만 해도 여러가지이다. 가장 최근 소식으로는 애플이 쿼타임 7에서 H.264를 지원하며, H.264가 차세대 HD-DVD의 표준이 될 것이라는 이야기가 있다. 엔드유저 입장에서 데스크탑 PC를 이용해 H.264를 체험해 보기 위해서는 배포되는 코덱을 설치해야 하는데 여러 코덱을 통합해놓은 패키지에는 보통 x264라는 GPL 라이선스 기반의 오픈코덱이 주로 포함되어 있고, 이번 기사에서도 x264 코덱을 사용해 테스트를 진행했다.

그 외에도 Nero Record2나, 쿼타임7, DVD Studio Pro, Sorenson Squeeze 4.1 for MPEG-4, VideoSoft H.264 Codec, Cyberlink PowerEncoder MPEG4 AVC Edition 등과 같은 상용 인코딩 소프트웨어와 코덱들이 여러 종류가 있다.

모바일 기기에서의 H.264 콘텐츠는 해상도와 비트레이트가 낮고 전용 프로세서를 통해 디코딩되기 때문에 CPU 점유율이 낮지만, 데스크탑에서 HD 콘텐츠를 재생할 때는 상당히 CPU 점유율이 높다. 앞의 코덱 소개에서도 밝혔듯 한번에 처리하는 블록의 사이즈가 1/4로 줄어들어 처리 횟수가 훨씬 늘어나기 때문이다. 때문에 테스트에는 Athlon64 x2 4800+ 프로세서를 사용했으며, 자신의 PC에서 재생이 가능할지 여부를 직접 확인해보시라는 의미에서 보드나라 자료실을 통해 인코딩 된 샘플 파일을 배포할 예정이다.



x264 코덱의 설정 메뉴이다. 비트레이트 기준과 품질 기준의 SinglePass 인코딩과 가변비트레이트 방식으로 압축하는 멀티패스 인코딩을 지원한다.



Advanced 옵션에 들어가면 무언가 많은 설정 옵션이 있지만, 코덱의 알고리즘에 정통한 사람이 아니라면 건드려 볼 엄두가 나지 않는다.

H.264 표준 기반 x264 코덱과 비교대상으로 삼은 코덱은 DivX 6와 WMV9 코덱이다. x264 코덱의 경우에는 임의의 특정 영상, 특정 해상도에서 에러를 출력하기도 하고, 일부 인코딩 프로그램에서 x264로 압축된 영상을 불러들여 편집할 수 없는 등 문제가 꽤 많아 체험해보기가 만만치 않았다. 이 점을 유념하고, 이번 테스트의 화질 샘플도 상용코덱과는 차이가 있을 수 있다는 점을 미리 밝힌다.

테스트는 가장 많이 사용될법한 비트레이트를 기준으로 진행하였다. 테스트 샘플은 영화 택시3

의 일부로 1440\*816 해상도의 8000kbps WMV9 영상이었으며 이를 Uncompressed RGB로 풀어 놓은 상태에서 낮은 비트레이트로 재인코딩 하였다. 해당 샘플은 마이크로소프트 홈페이지의 WMV9 관련 페이지에서 다운로드 받을 수 있다.

비트레이트	초당 용량	시간당 용량
150 kbps	18.75 KB /s	65.92 MB
800 kbps	100 KB/s	351.56 MB
1500 kbps	187.5 KB/s	660 MB
3000 kbps	375 KB/s	1318.35 MB

원본 해상도를 유지한 채 800K, 1500K, 3000K의 세가지 주요 비트레이트와 휴대용기기에서 주로 사용하는 320\*240 / 150Kbps를 비교하였다.

4. 화질비교 800K

원본의 해상력을 흉내내기에는 800Kbps는 비트레이트가 많이 부족하다. H.264 코덱으로 인코딩 된 결과물의 특성은 깎두기 픽셀이 생기지 않는다는 점이다. 눈이 튀어오른 부분을 보면 원본에 비해 흐리게 표현 되었지만, 같은 비트레이트에서 DivX6나 WMV9은 깎두기 픽셀이 많이 생기는 것에 비하면 액정 크기가 제한된 모바일 기기에서는 비교적 보기 좋은 결과물을 보여줄 것으로 예상해볼 수 있다. 눈발의 표현은 많이 뭉개졌지만, 번호판의 가독성은 다른 두 코덱에 비해 나쁘지 않은 수준이다.



낮은 비트레이트는 사람 얼굴을 위와 같이 괴물로 만들어버릴 수도 있다. H.264는 이번에도 대체적으로 흐리게 처리하는 모습을 보여줬다. 얼굴의 경우 원본에 비해서는 많이 흐려져 보기 좋지 않지만, DivX6의 결과물보다는 낫다. 이번 샘플에서는 비록 깎두기는 졌지만 WMV9이 가장 얼굴의 디테일을 잘 살리고 있다. 얼굴 오른쪽에 사선이 있는데 사선의 디테일은 H.264가 가장 우수하다.



굉장히 빠르게 움직이는 장면인데, 여기서 DivX6의 1-Pass CBR 결과물이 상당히 좋지 못하다. 2-Pass VBR 인코딩에서는 좋은 결과를 보여주는 DivX6지만, 때에 따라서 치명적인 약점이 있는 것이다. 대신 배경인 산의 디테일에서는 DivX6가 H.264보다 좀 더 좋았고, WMV9는 평균적으로 좋은 모습이긴 하지만 블럭들이 생기는 것은 피할 수 없었다.





800Kbps로 인코딩 된 H.264 코덱의 결과물과 3000Kbps의 높은 비트레이트로 인코딩된 결과물을 비교해 보아도 H.264의 결과물은 손색이 없다. 샤프한 맛은 없지만 뭉개지면서 생기는 패턴이나 격자무늬가 훨씬 부드럽고 자연스럽게 처리 되는 것을 볼 수 있다. 데스크탑에서는 저런 뭉개짐이 거슬릴 수 있겠지만, 모바일 기기용 저화소 액정에서는 보다 자연스럽게 표현된다.

##### 5. 화질비교 1500/3000K

1500kbps는 인터넷에 릴리즈 되는 영화 한편당 CD 1장 분량의 고화질 영상물의 비트레이트와 비슷하며, DMB 기기의 최대 대역폭인 2048Kbps 이내에 들어온다. 음성 부분의 용량을 감안한다면 H.264 코덱을 통해 스트리밍 되는 고화질 DMB 콘텐츠는 1500kbps를 넘지 않는 선일 것이다. 1500kbps에서는 1 Pass CBR 및 2 Pass VBR의 결과물을 비교해 보았다. 압축률을 보는 것은 1 Pass CBR만으로 충분하지만, 모션의 변화가 급격한 화면에 비트레이트를 적절히 할당하는 기술도 코덱의 능력이기때문에 VBR 테스트를 포함한 것이다.



원본 소스가 DV 소스가 아닌 WMV9 8000kbps로 압축되어 있던걸 폰 영상이기 때문에 원본에도 약간의 화질손상이 있는 상태이다. 스키 타는 여인보다는 나무의 디테일을 잘 살펴보도록 하자.





H.264로 인코딩 된 결과물은 대체적으로 비트레이트를 가리지 않고 뭉개지는 것이 특징이다.

(물론 상용화된 코덱과는 차이가 있을 수 있다) 때문에 애니메이션처럼 면과 면의 외곽을 표현하는 능력에서는 유리하지만 풍경 등의 묘사에서는 다소 떨어질 수 있다. 2-Pass 인코딩 되었을 때 화질 향상이 가장 좋았던 코덱은 WMV9이지만 좋아진 상태에서도 깎두기 난무는 피할 길이 없었고, DivX6는 나무의 디테일이 약간 좋아졌다.



2-PASS 인코딩 영상을 줄여 놓은 것인데, 확대 해 놓은 스틸컷으로는 어느 쪽이 보기 좋은지를 판단하기 힘들었지만, 위의 영상을 보면 H.264가 흐릿하다는 느낌을 바로 느낄 수 있다. 이번에는 3000Kbps 품질을 보자.





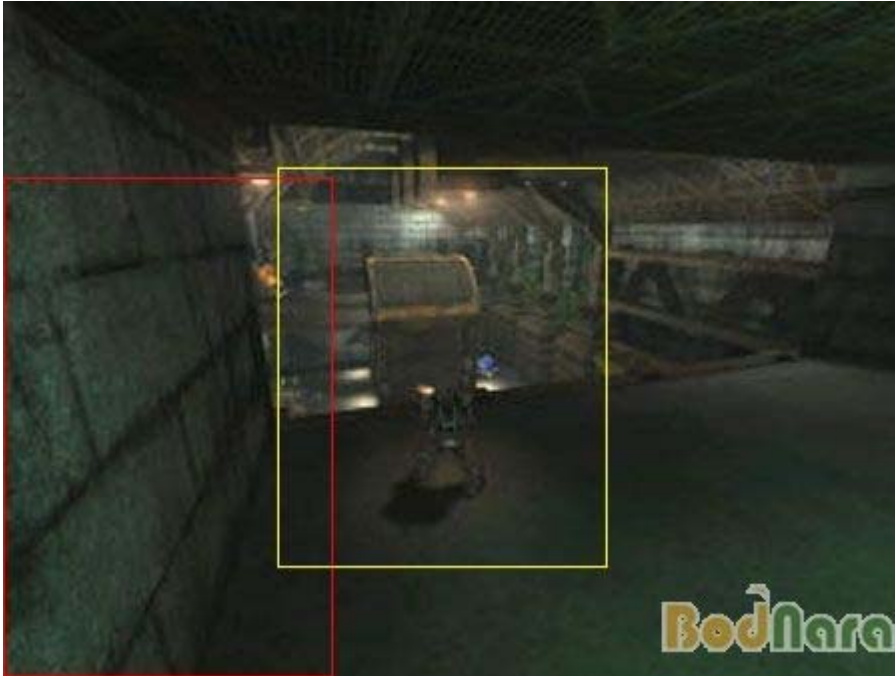


3000k로 비트레이트가 높아져도 H.264는 더 이상의 뚜렷한 화질 개선을 찾기 어렵다. 아무래도 x264 코덱으로는 HD 해상도의 영상을 담기에는 DivX6나 WMV9에 비해 그리 메리트가 없어 보인다.

#### 6. 화질비교 - 저해상도 150K

휴대용 기기에서 지원하는 동영상의 규격은 대부분 320\*240 ~ 640\*480 수준이기 때문에 HD

급의 해상도는 상용화된 H.264 콘텐츠와는 거리가 있다. 이번에는 해상도를 320\*240으로 놓고, 150kbps 수준의 낮은 비트레이트로 압축해 보았다. 이 해상도와 비트레이트는 PDA에서 흔히 사용되는 수준이다.



이번에는 소스로 언리얼토너먼트 2003의 화면을 Fraps로 캡처한 영상을 320\*240으로 리사이즈한 Uncompressed RGB 파일을 사용하였다. 앞서 사용했던 1440\*816인 파일보다 화소수가 1/15 정도에 불과해 150kbps라 하더라도 비트레이트가 상대적으로 높지만, 압축을 위해 채취하는 샘플 블록의 크기가 전체 해상도에서 차지하는 비중이 크기 때문에 결과가 전혀 달라질 수 있다.



저해상도로 인코딩된 결과물을 비교해본 결과 DivX6가 가장 우수했고, H.264는 이번에도 역시 다소 흐린 모습이다. WMV9은 항상 생기는 깎두기가 여전히 심각한 수준이었다. H.264가 안티알리아싱을 적용시킨듯한 외각선을 표현해준다면, DivX6는 비등방성 필터링을 적용한듯한 벽면을 표현해 주고 있다.



벽면의 디테일에서는 H.264의 결과가 가장 안 좋았지만, 사람 주변의 그림자를 표현하는 능력은 H.264가 가장 좋다. 대신 캐릭터 및 배경 묘사가 상당히 흐리다.







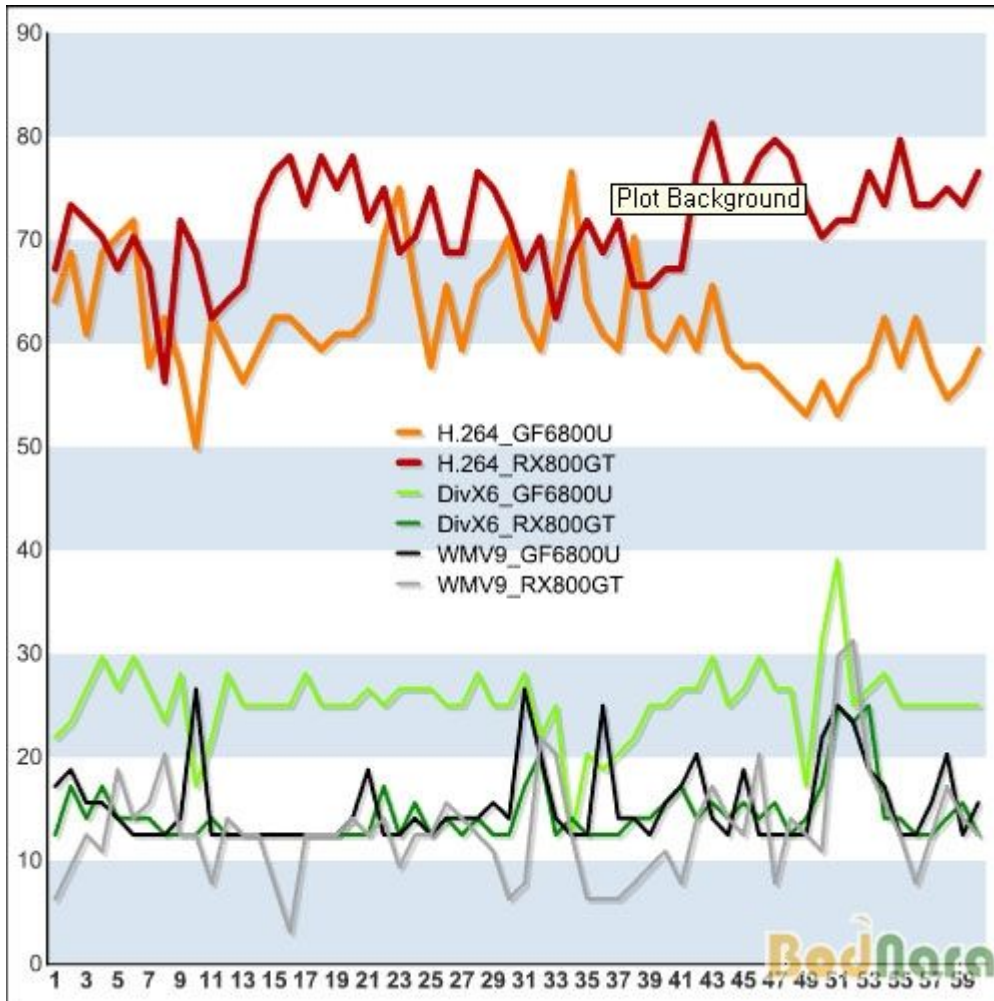
위의 두 영상 모두 H.264 코덱의 약점이 드러난다. 고해상도로 인코딩 된 결과물에서 흐려지는 현상은 재생할 액정의 해상도가 낮을 때는 숨겨지는 부분이지만, 해상도를 낮춰 인코딩할 경우 결과물이 그리 좋지 못하다.

#### 7. CPU 점유율 테스트

지난 Athlon64 x2 프로세서의 테크닉 기사에서 H.264 코덱으로 인코딩 된 결과물을 재생할 때 차지하는 CPU 점유율에 대해 다뤄진 적이 있는데, H.264 코덱은 최신 고클럭 CPU에서도 상당히 높은 점유율을 차지하며, 듀얼코어에서도 만만치 않은 CPU 점유율을 보인다.

작업관리자의 CPU 점유율 그래프를 보여드리기 위해 필자가 기사 작성용으로 사용하는 Athlon XP 2500+ 시스템에서 H.264 코덱으로 인코딩 된 1440\*816 사이즈 동영상을 재생하고 그래프를 캡처 하려고 했다. 하지만 재생 순간부터 CPU 점유율이 곧바로 100%로 치솟으며 작업관리자의 그래프가 그려지지 않았고 동영상만 재생되고 다른 모든 작업은 멈춰버렸다. 따라서 웬만한 고사양 시스템이 아닌 이상에는 H.264 코덱을 다른 하드웨어 가속 지원 없이 사용하기는 어려워 보인다.

다음은 Athlon64 x2 4800+ 시스템에서 측정한 CPU 점유율이다. 자신의 시스템에 H.264 코덱을 사용할 수 있을지 여부는 자료실에 인코딩 결과물 샘플을 올려드릴 예정이니 직접 확인해보시기 바란다.



	nVIDIA Geforce 6800 Ultra	ATI Radeon X800 GT
	평균 CPU 점유율	
H.264	61.6	71.8
DivX6	25.3	14.5
WMV9	15.4	12.9
	Peak CPU 점유율	
H.264	76.6	81.3
DivX6	39.1	25.0
WMV9	26.6	31.3

그래프를 보면 한 눈에 알 수 있듯, 현존하는 가장 빠른 데스크탑 프로세서 중 하나인 Athlon64 x2 4800+에서도 그래픽카드에 따라 순간 CPU 점유율이 80%를 넘어간다. 다른 코덱들에 비해 최대 6배에 가까운 프로세싱 능력이 필요한 것이다. 1440\*816짜리가 이 정도이니, 1920\*1080의 풀 HD 해상도 영상은 어떠할지 겁날 지경이다. 게임 F.E.A.R만 업그레이드의 공포를 주는 것이 아니라 H.264도 또 하나의 공포인 것이다.

또 한가지 재미있는 결과는 nVIDIA Geforce6 시리즈와 ATI의 Radeon X800 시리즈의 CPU 점유율이 판이하게 다르게 나왔다는 점이다. nVIDIA는 H.264의 하드웨어 가속을 지원하겠다고 밝혔던 바 있고, ATI는 MPEG-4 영상의 하드웨어 가속을 지원한다고 했던 것의 차이가 여기서 드러난다. ATI가 R5xx 계열을 출시하면서 H.264 하드웨어 가속을 지원하겠다고 밝혔지만, 아직까지는 nVIDIA Geforce 6 계열 그래픽카드가 H.264 코덱의 재생에 더 유리하다.

#### 8. 데스크탑을 업그레이드할 것인가? 아니면?

서로 다른 코덱으로 압축한 결과물의 화질을 동일하게 맞출 수 없기 때문에 코덱의 압축률의 순위를 객관적으로 매길 수 있는 척도는 없다. 그렇기 때문에 H.264 코덱의 압축률이 좋더라고 표현하는 것이 객관적인 결론이라고 단언할 수 없다. 특히 스틸샷으로 일부 화면만 보여줬을 때는 더더욱 어떤 코덱으로 압축했을 때 화질 손상이 적은지를 판단하기 어렵다.

필자의 주관으로 판단한 H.264의 특성은 낮은 비트레이트에서의 상대적인 화질 손상이 적다는 점, 비트레이트가 높아진다고 해서 획기적으로 화질이 좋아지지 않는다는 점(2-Pass VBR의 화질 차이가 적은 것도 같은 맥락이다. 특정 프레임에 더 높은 비트레이트율을 준다고 해도 크게 달라지지 않기 때문에..), 그리고 해상도가 낮은 소스를 압축했을 경우에는 뭉개짐이 특히 심하다는 점 등이 있겠다.

여기서 한 가지 의문이 생기는데, 비트레이트가 높아져도 화질 개선이 뛰어나지 않고 오히려 다른 코덱보다 뭉개지는 현상이 생긴다면 차세대 HD-DVD 포맷으로 적합하지 않은 것이 아니냐는 점이다. 이 부분에 대해서 역시 주관적인 느낌을 말하자면 부분을 확대한 정지화면이 아닌 움직이는 동영상일 경우에는 Divx와 WMV보다 H.264 코덱이 더 자연스러운 화면을 보여 주었다. 뭉개지는 범위는 원본 해상도가 높아질수록 좁아지기 때문에 화면 해상도에 맞춰 리사이즈된 화면에서는 격자무늬가 생기는 것보다는 뭉개진 편이 보기가 더 좋다. 해상도가 높아지면 그만큼 많은 비트레이트를 필요로 하기 때문에 낭비되지 않을만큼 소스 해상도에 적당한 비트레이트를 부여한다면 고해상도 영상의 저장 포맷으로도 손색이 없을 것이다.

그렇지만 H.264 코덱이 진정 잘 어울리는 부분은 역시 DMB와 같은 모바일 기기이다. H.264 코덱이 낮은 비트레이트일때 특히 다른 포맷에 비해 좋은 화질을 보여 주었기 때문에, 네트워크 대역폭이 1~2Mbps 정도로 제약된 환경에서 SD급 화상을 전송하기에는 이번에 테스트 해본 3가지 코덱 중에는 가장 효율적이다. 특히 휴대용기기의 액정 해상도가 낮기 때문에 전송되는 소스의 해상도가 더 높을 경우 뭉개짐이 가려지고 자연스러운 화면을 보여주게 될 것이다.

데스크탑에서의 CPU 점유율도 문제가 되는데, 앞으로 HD-DVD를 감상하기 위해서는 듀얼코어 프로세서 또는 H.264 하드웨어 가속이 지원되는 그래픽카드는 필수적이 될 것이다. 물론 이 문제는 현재 HDTV 수신카드가 DxVA 가속을 지원하는 그래픽카드를 필요로 했던 문제처럼, 시간이 지나면 모든 하드웨어가 HD-DVD에 적합한 방향으로 발전할 것이기 때문에 소비자가 부품을 고를 때 머리 아파할 일은 없게 될 것이다. 물론 지금 당장 고해상도 H.264 콘텐츠를 보려 한다면 좌절하게 될테지만...